Spark 框架下利用分布式 NBC 的大数据文本分类方法 *

臧艳辉 [1]，赵雪章 [1]，席运江 [2]

(1. 佛山职业技术学院，广东 佛山 528137; 2. 华南理工大学，广州 510000)

摘　要： 针对现有面向大数据的计算框架在可扩展性机器学习研究中面临的挑战，提出了基于 MapReduce 和 Apache Spark 框架的分布式朴素贝叶斯文本分类方法。提出的方法通过研究 MapReduce 和 Apache Spark 框架的适应性来探索朴素贝叶斯分类器（NBC），并研究了现有面向大数据的计算框架。首先，基于朴素贝叶斯文本分类模型将训练样本数据集分为 m 类。进一步在训练阶段中，将前一个 MapReduce 的输出作为后一个 MapReduce 的输入，采用四个 MapReduce 作业得出模型。该设计过程充分利用了 MapReduce 的并行优势。最后，在分类器测试时取出最大值所属的类标签值。在 Newgroups 数据集进行实验，在所有五类新闻数据组上的分类都取得了 99%以上的结果，并且均高于对比算法，证明了本文方法的准确性。

关键词： 文本分类；MapReduce；Spark 框架；分布式；朴素贝叶斯分类器；机器学习

中图分类号： TP311　　doi: 10.3969/j.issn.1001-3695.2018.07.0407

Text classification of big data using distributed naive Bayesian classifier under spark framework

Zang Yanhui[1], zhao Xuezhang[1], Xi Yunjiang[2]

(1. *Foshan polytechnic*, *Foshan Guangdong* 528137, *China*; 2. *School of Economics & Management*, *Guangzhou* 510641, *China*)

Abstract: Aiming at the challenges faced by the existing big data-oriented computing framework in the study of extensible machine learning, a distributed naive Bayesian text classification method based on MapReduce and Apache Spark framework is proposed. The proposed method explores the Bayesian network classifier by studying the adaptability of MapReduce and Apache Spark frameworks, and studies the existing computing framework for big data. First, the training sample data set is divided into m classes based on the naive Bayes text classification model. In the training phase, the output of the previous MapReduce is used as the input of the next MapReduce, and four MapReduce jobs are used to derive the model. This design process makes full use of the parallel advantages of MapReduce. Finally, when the classifier is tested, the value of the class label to which the maximum value belongs is fetched. Experiments in the new group's dataset have resulted in more than 99% of the results on all five types of news data sets, and are all higher than the comparison algorithm. Proved the accuracy of the method of this article.

Key words: text classification; MapReduce; Spark framework; distributed; naive Bayes classifier (NBC); machine learning

0　引言

可扩展性一直是数据挖掘和机器学习领域的一个重要的研究领域[1~4]。为了解决更大和更复杂的问题，算法不断得到改进以扩大规模。随着硬件的持续发展以及技术地不断改进，研究者面临着越来越多来自新兴领域的挑战。近年来，互联网经济的崛起以及支付方式的转变，为传统商业带来了新的发展空间。商业的进一步繁荣带来了存储海量数据的需求。这种通常被称为大数据的新数据现象为机器学习打开了一扇新窗口[5~7]。

这些新问题的解决需要新颖的解决方案，为此，诞生了各种新型的适用于高度分布式和面向数据的新计算框架。在现有的提案中，MapReduce 获得了巨大的认可，成为处理海量数据集的标准[8]。自从 MapReduce 开始以来，MapReduce 在机器学习解决方案方面一直处于积极发展的阶段，其中特定技术在适应框架方面获得了更多的普及。在文献[9]中提出了一种由多个分布式阶段组成的高效且可扩展的 kNN 查询模型。Apache Hadoop 是作为 MapReduce 的标准开源实现而建立的，然而 Apache Spark 已经成功地提出了改进的数据抽象，基于主内存

收稿日期： 2018-07-02；**修回日期：** 2018-08-27　**基金项目：** 国家自然科学基金资助项目（71371077）；佛山市科技计划项目（2015AB004241）

作者简介： 臧艳辉（1978-），女，湖北襄阳人，高级工程师，硕士，主要研究方向为大数据、图形图像处理（zangshirley@126.com）；赵雪章（1972-），男，河南南阳人，副教授，硕士，主要研究方向为大数据、智能算法；席运江（1973-），男-，河南南阳人，副教授，博士，主要研究方向为知识网络、大数据等.

的更快的执行环境以及友好的编程接口。在文献[10]中提出了基于 Hadoop 框架的树增广的朴素贝叶斯分类方法，将训练数据集分成不同的块，然后将它们分布在可用的计算节点上，并且取得了较高的分类结果。然而该方法计算成本较高，难以广泛推广。

本文提出了一种基于 MapReduce 和 Apache Spark 框架的分布式朴素贝叶斯文本分类方法。提出的方法侧重于这些模型的学习阶段，为此本文介绍了一个通用框架。该框架描述了 MapReduce 范例下朴素贝叶斯分类器(naive Bayes Classifier, NBC)的完整系列。此外，还从理论和实践两个方面讨论了这些模型的最佳特性和主要缺陷，其目的在于找出未来的研究路线，以专门设计的算法来解决这个问题。在 Newgroups 数据集进行实验，取得了 99%以上的结果，并且均高于对比算法。证明了本文方法的准确性。

1 MapReduce 和 Apache Spark 框架

1.1 MapReduce 框架

MapReduce 编程范例是由 Google 于 2003 年设计的大数据横向扩展数据处理工具[11~13]。它被认为是互联网上最强大的搜索引擎，迅速成为用于通用数据并行化的最有效技术之一。图 1 为 MapReduce 数据流程。

MapReduce 基于两个单独的用户定义的基元：Map 和 Reduce。Map 函数以键值(＜*key*，*value*＞)对的形式读取原始数据，并将它们转换为一组中间的(＜*key*，*value*＞)对，可能是不同类型的。键和值类型都必须由用户定义。然后，MapReduce 将与同一中间键关联的所有值合并为列表(混洗阶段)。最后，Reduce 函数从映射中获取分组输出并将其聚合为一组较小的对。这个过程可以如图 1 所示进行模式化。这个透明且可扩展的平台自动处理分布式集群中的数据，从而减少用户的技术细节，如数据分区、容错或作业通信。

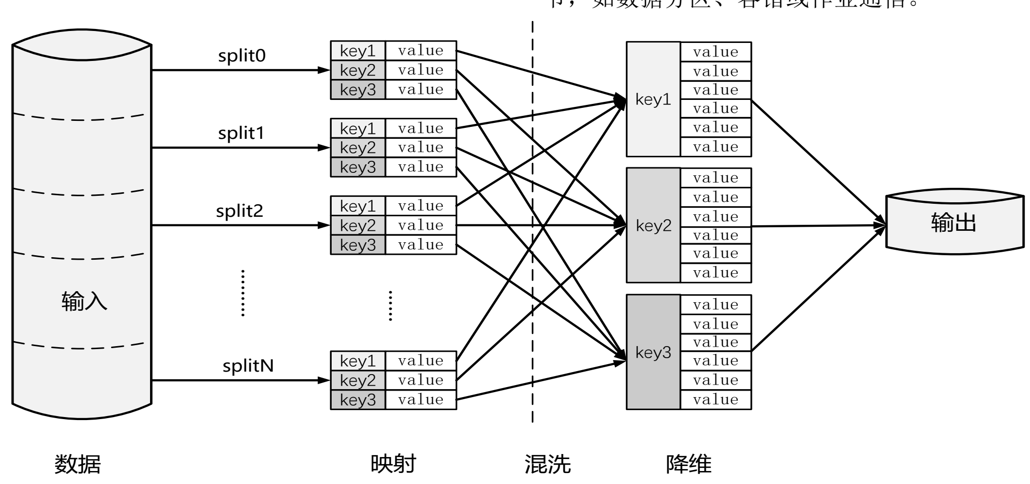


图 1　MapReduce 数据流程

MapReduce 的主要潜力是它提出的计算抽象，其中整个处理被划分成更小的任务类型 Map 和 Reduce，沿着集群均匀分布和处理。从业者只需负责提供这两个功能，避免将处理适配到群集的底层架构或数据的性质。该框架为并行数据处理提供了一个高度可扩展的容错环境。MapReduce 过程遵循两步过程，其中集群的体系结构以主/从方案进行组织：一个主节点配置作业，并将计算任务和输入数据分布在将执行处理的工作节点集合上。首先，输入数据被分割成较小的分区或块，它们分布在工作节点中，作为给定数量的 Map 任务的输入。

1.2 Apache Spark 框架

Apache Hadoop 是 MapReduce 在商业集群上大规模处理和存储的最流行的开源实现[14~16]。由于其性能、开源特性、安装设施和分布式文件系统(Hadoop 分布式文件系统，HDFS)，该框架在许多领域的应用已经非常普遍。尽管 Hadoop 和 MapReduce 非常受欢迎，但在许多情况下，如在线或迭代计算，都显示出不适合。无法通过内存原语重用数据，这使得 Hadoop 在很多机器学习算法中的应用变得不可行。

Apache Spark 是一种用于大规模数据处理的新颖解决方案，被认为能够解决 Hadoop 的缺点。Spark 是作为 Hadoop 生态系统的一部分引入的，它旨在与 Hadoop 合作，特别是通过使用其分布式文件系统。该框架提出了一套超越标准 MapReduce 的内存原语，目的是在分布式环境中更快速地处理数据，速度比 Hadoop 快 100 倍。

Spark 是一种基于弹性的分布式数据集(resilient distributed datasets，RDD)[17]，这是一种用于以透明方式并行计算的特殊类型的数据结构。这些并行结构让人们坚持并重用结果，并将其缓存在内存中。此外，还可以管理分区来优化数据放置，并使用大量透明基元操作数据。所有这些功能都允许用户轻松设计新的数据处理管线。

可扩展的机器学习库(MLlib)建立在 Spark 之上，这要归功于其对迭代过程的隐含适用性。当前版本的 MLlib(v1.6.0)包含大量的标准学习算法和统计工具，涵盖了知识发现过程中的许多重要领域，如分类、回归、聚类、优化或数据预处理。 MLlib 是 MLbase 平台的关键组件。它提供了一个高级 API，使用户更容易连接多个机器学习算法。但是这个平台并不包含贪婪学习算法，如 kNN 算法。

更具体地说，除了机器学习中大多数 MapReduce 自适应算法所遵循的水平策略外，本文还定义了一种基本的并行计算垂直化方法。这种垂直策略可以通过智能程序进一步扩展，以平衡不同垂直 Map 任务之间的数据复制，从而优化高维域算法的可伸缩性。

根据最先进的 Apache Spark 计算框架，已经提出了此框架的特定实现。该软件已经过大量不同的问题基准测试，并讨论了许多性能指标。除了此基准测试之外，还测试了计算体系结构的几种配置，以提供本文提议的可伸缩性属性的总体概述。图 2 展示了 Spark 运行过程中 RDD 中的数据转换和操作。

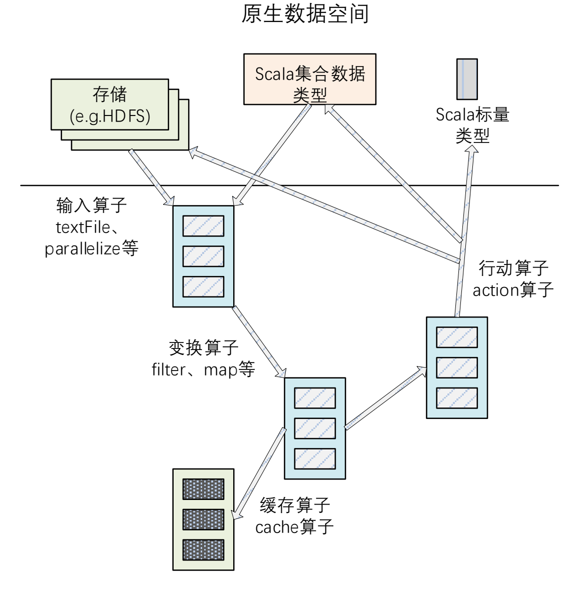


图 2　Spark 运行过程中 RDD 中的数据转换和操作

2　贝叶斯网络分类器

2.1　符号表示

给定离散预测随机变量的向量 $A=\{X_1\ldots,X_n\}$ 和类变量 C，$c\in\Omega_C=\{c_1,\ldots,c_l\}$，本文希望从包含 m 个标签化示例 $\{(x^{(1)},c^{(1)}),\ldots,(x^{(m)},c^{(m)})\}$ 的训练集 $\mathcal{D}$ 中推导出一个模型，其中 $x^{(j)}=(x_{j1},\ldots,x_{jn})$。表 1 列出了本研究中使用的不同符号注释。

表 1　本文中使用的不同符号元素的总结

$\mathcal{D}$	训练集
$\mathcal{D}_k$	训练数据的一个子集
n	预测器属性的数量
m	训练数据中示例数量
A	预测特征或属性的集合
C	类变量
$\bar{v}$	每个属性对应值的平均数量
$\Omega_C=\{c_1,\ldots,c_l\}$	类变量的域
I	类的数量
X_i	X 中第 i 个属性的值
$pa(X_i)$	给定的 BN 上 X_i 的一组父代

2.2　贝叶斯网络分类器

在其他受欢迎的监督分类模型中，近年来基于贝叶斯网络的分类模型越来越流行。从概率方法来看，希望给定一个例子 X 来估计 $p(C|X)$，且 $C\in\Omega_C$。贝叶斯分类器将选择最大化后验概率(MAP)的值 c^k：

$$c^k=\arg\max_c p(c|x)=\arg\max_c p(x,c) \tag{1}$$

贝叶斯网络(Bayesian network, BN)是一个概率图形模型，可以让本文有效地表示和操纵概率分布。它由两个部分定义：它由两个部分定义：一个是由有向无环图 (V,E) 表示的图形结构 $\mathcal{G}$，其中 V 是一组表示 A 中变量的节点，E 是一个边集合，用来编码节点之间的依赖关系；以及一组数字参数 Θ，用来编码关于图中编码的变量和依赖关系的定量信息。具体地，对于图 $pa(X_i)$ 中的每个变量 $X_i\in A$ 及其父集、条件概率分布表(CPT) $p(X_i|pa(X_i))$ 被存储。这种由 BN 编码的表示可以用于恢复由于马尔可夫规则而导致的联合概率分布：

$$p(X_1,\ldots,X_n)=\prod_{i=1}^{n}p(X_i|pa(X_i)) \tag{2}$$

由于贝叶斯网络形式具有良好的数学基础和预测模型的能力，所以被广泛用于人工智能领域的许多任务。然而一般的 BN 模型已证明对监督分类问题不具有竞争力。出于这个原因，已经提出了具体的模型，其中 BN 的结构适合于处理这种特定情况，通常将类变量视为具有更重要的依赖性角色的节点。这种模型通常被称为贝叶斯网络分类器(BNC)，并被认为是许多领域的最先进方法。

也许最常见的 BNC 模型是流行的朴素贝叶斯(NB)分类器，其中 A 中的所有预测特征被认为是独立于 C 类的。尽管这是一个很强的假设，在实际数据中很少成立。NB 分类器的性能已经被证明对许多问题具有竞争力，并且由于其高效的训练是许多从业人员的合适人选。这种独立性假设为模型定义了一个固定的结构，所以没有必要从训练数据中引入相关性。

训练 NB 分类器并根据数据估计其参数，其中使用的最大似然估计(MLE)算法，以及诸如拉普拉斯的平滑策略具有 $\mathcal{O}(nm)$ 的计算复杂度和 $\mathcal{O}(nl\bar{v})$ 的空间复杂度，这也是所得模型的空间复杂性。

2.2.1 树增广的朴素贝叶斯

树增广模型允许每个预测属性依赖于除了该类外的额外变量来放宽条件独立假设。为了选择这种特定的相关性，提出了一种结构学习算法，通过条件互信息(CMI)来构建最大生成树：

$$MI(X_i,X_j|C)=\sum_{c\in\Omega_C}\sum_{x_i\in\Omega_{X_i}}\sum_{x_j\in\Omega_{X_j}} p(x_i,x_j,c)log(\frac{p(x_i,x_j|c)}{p(x_i|c)p(x_j|c)}) \tag{3}$$

一旦获得树，就通过任意选择一个根节点并按照拓扑顺序定位边来构建 DAG；最后，类变量作为公共的父项添加到所有节点。该算法需要 $\mathcal{O}(mn^2)$ 的计算复杂度来计算每对属性的 CMI，为此构建一个空间复杂度为 $\mathcal{O}(ln^2\bar{v}^2)$ 的三维表。构建最大生成树的计算复杂度为 $\mathcal{O}(n^2\log n)$。模型的参数化学习可以

计算具有 $\mathcal{O}(n\bar{v}l)$ 复杂度的所需概率表；得到的 TAN 分类器需要空间复杂度为 $\mathcal{O}(ln\bar{v}^2)$ 的存储。

2.2.2 k-依赖估计量

k 依赖估计可以看做是前面想法的推广，其中每个预测变量的模型中允许有多个 k 个附加父项。该算法可以探索更广泛的分类器，从 k 开始，如果 $k=0$，随着 k 的增加，朝向全 BN 模型。分类器的结构通过三步算法学习，该算法也依赖于互信息：

在预测属性之间通过它们的互信息与类别变量 $MI(X,C)$ 构建排名 σ。

$$MI(X_i|\ C)=\sum_{c\in\Omega_C}\sum_{x_i\in\Omega_{X_i}}p(x_i,c)log(\frac{p(x_i,c)}{p(x_i)p(c)})\ (4)$$

对于每个属性 X_i，本文计算条件互信息 $MI(\bullet,X_i|C)$，给出了在 σ 之前的属性子集的类别：$\{X_1,\ldots,X_i-1\}$。然后将依赖性最高的最佳 k 个属性作为 X_i 的父项。

最后，类变量 C 被添加为所有预测属性的父项。

学习 kBD 分类器的网络结构具有 $\mathcal{O}(n^2m)$ 的计算复杂度和 $\mathcal{O}(l\bar{v}n^2)$ 的空间复杂度，因为它还构建三维表格。计算该特定网络的 CPT 需要 $\mathcal{O}(n(m+\bar{v}^k))$，并且需要空间复杂度为 $\mathcal{O}(kn\bar{v}^kl)$。

2.2.3 平均 k 依赖估计量

平均 k-依赖估计量(AkDE)背后的思想与以前的略有不同，而不是从 NB 分类器学习增广结构，通过定义固定结构简单模型的集合来避免这种代价高昂的操作。其中，对于每个模型，所有属性都依赖于类以及被称为超级父亲的附加属性。对于 $k=1$ 的值 AkDE 分类器由 n 个 SPODE 组成，每个 SPODE 具有不同的作为超父亲的 $X\in A$ 属性。分类是通过平均集合中模型的单独预测而获得的。

学习一个 AkDE 分类器需要计算一个 $k+2$ 维表来估计所需的参数，计算复杂度为 $\mathcal{O}(mn^k)$，空间复杂度为 $\mathcal{O}(\binom{n}{k}l\bar{v}^{(k+1)})$。

3 文本分类

3.1 分布式朴素贝叶斯网络分类器

根据之前的分析，所描述的算法可以找到一个共同的问题：学习过程的主要计算负担(结构或参数)是由训练数据计算多维列联表造成的。如果本文使用 MLE 等频率方法，所描述的度量 MI 和 CMI 都需要以与 BN 模型参数学习相同的方式来估计概率分布。计算这些列联表是复杂度为 $\mathcal{O}(mn^k)$ 的算法中要求最高的部分，其中 m 对应于读取完整训练数据，并且 $k+1$ 与这样的表的维度相对应，即估计 NB 分类器的参数需要学习涉及每个属性($k=1$)和类的二维表，而对于 A1DE 分类器，需要三维表，涉及每对属性($k=2$)和类。

一般来说，这些列联表可以通过给定属性集合 $S\subseteq A\cup\{C\}$ 的不同配置中的直方图或频率计数分布来表示。本文将计算子集 $S=\{X_1,\ldots,X_r,C\}$ 中属性的联合状态集的每个出现的频率 $\otimes\Omega_{X_i}\setminus X_i\in S$，其中，列联表 $\#_{\mathcal{D}}(S)$ 将 $\otimes\Omega_{X_i}$ 中定义的每个可能配置的出现次数存储在给定数据集 $\mathcal{D}$ 中。

学习一种特定类型的 BNC 需要计算给定数量的列联表，这些列表是为一组变量 $\Psi=\{S_1,\ldots,S_v\}$ 定义的。例如学习朴素贝叶斯分类器需要估计每个属性和类的计数，因此估计计数的变量集将为 $\Psi=\{S_i=\{X,C\}\forall X\in A\}$。 在以前的工作中，提出了用于学习特定模型的 MapReduce 算法。然而，为了定义覆盖所有提议的 BNC 算法的通用框架，可以建立一个通用模式。从现在开始，本文将以这种普遍的观点提出本文的问题，目的是定义一个通用的程序来学习任何描述的模型；稍后，本文将能够实例化此框架以匹配特定的算法。该通用程序策略基于计算给定的一组属性组合 Ψ 的列联表，该列表将根据所选择的特定模型进行设置。

计算大量数据的这种列联表格是 MapReduce 范例中自然存在的问题。一般来说，本文可以确定两种不同的策略来定义一个并行方案来获得给定一组属性/类组合的计数，其中每个策略都旨在并行化可伸缩性问题的不同组件。

水平并行：首先将训练数据集分成不同的块，然后将它们分布在可用的计算节点上。计算节点对每个数据块和属性组合 $S\in\Psi$ 部分计算列联表。最后，可以收集和汇总部分分布以恢复整个数据集的全部分布。

垂直平行：旨在分配 Ψ 中不同子集的计算。该策略适用于存在大量属性的高维域，从而使模型的大小随着多项式复杂度而增长。在这些情况下，数据集的简单水平分布意味着在收集不同的部分计数时会产生大量开销，因此，即使与前一种方法结合使用，也可以使用垂直并行性。这是通过将每个应急表计算为一个单独的并行任务来完成的。该策略涉及根据不同子集 $S\in\Psi$ 的数据集的垂直分布；并且由于子集可能重叠，所以采用这种复杂的组合问题以最小化数据复制，并确保跨不同节点的计算平衡。

3.2 朴素贝叶斯文本分类模型

根据之前的定义来估计训练数据集 $\mathcal{D}$ 上的特定数量的属性子集 $\Psi=\{S_1,\ldots,S_v\}$ 的列联表。水平策略通过将数据划分为块 $\{\mathcal{D}_1,\ldots,\mathcal{D}_h\}|\ h\le m$，理想情况下它将分配给 h 个 Map 任务。这些任务中的每一个计算可用本地块 $\mathcal{D}_i$，以及每个子集 $S_j\in\Psi$ 的部分列联表 $\#_i(S_j)$。 然后，每个映射任务为每个子集发出一组键值对 $\langle S_j,\#_i(S_j)\rangle$，并将它们与它们的相应分布相关联。

在 Reduce 阶段，这些对将按照它们的键(代表属性子集)进行分组并发送到相应的 Reduce 任务，其中理想的任务数为 v，每个子集一个。Reduce 阶段并行聚合不同属性子集的部分分布，并发出包含相应子集及其为完整数据集计算的完整列联表的新键—值对。通过为 Ψ 提供特定范围的属性子集，以前的框架可以实例化到任何描述的 BNS 中。

将训练样本数据集分为 m 类，记为 $C=\{C_1,C_2,\ldots,C_m\}$。而事件 C_j 类发生的先验概率则用 $P(C_j)$ 表示，，且 $P(C_j)>0$，其属于 C_j 类的条件概率是 $P(di|C_j)$。那么，对于任一新文档：

$$di=\left(w_1,w_2,\dots,w_{|V|}\right),i=1,2,\dots,l \tag{5}$$

其中：w_k 为特征词，$k=1,2,\dots,|V|$；则 $|V|$ 表示特征词的总个数；V 表示特征词的集合。

贝叶斯公式计算的 C_j 类的后验概率表示为

$$P\left(C_j\mid di\right)=\frac{P\left(C_j\right)P\left(di\mid C_j\right)}{\sum_{k=1}^{|V|}P\left(C_j\right)P\left(w_k|C_j\right)} \tag{6}$$

4　实验评估

本文针对现有方法在可扩展性机器学习研究中面临的挑战，提出了基于 MapReduce 和 Apache Spark 框架的分布式朴素贝叶斯文本分类方法。提出的方法关注监督分类问题，通过研究 MapReduce 和 Apache Spark 框架的适应性来探索贝叶斯网络分类器。为了证明提出方法的有效性，进行了实验验证。

4.1　运行环境

本文将使用由一个主机和六个从节点组成的计算机集群，每个节点配备双 Intel Xeon E5-2609v3 1.90 GHz 六核处理器和 64 GB RAM。每个工作节点都在 4x1 TB 磁盘上运行 HDFS 文件系统，并由 Cloudera CDH 5.5 分发管理。

在独立部署中，Spark 1.6.0 中选择 MapReduce 环境。本文将通过提供不同数量的资源来启动集群的不同配置，以测试不同架构布局的算法行为。

4.2　分类器训练与测试

本文采用四个 MapReduce 作业得出模型。需要设置多维列联表的拓扑结构，通过正确识别 Ψ 来计算。 在这种情况下，NB 分类器对每个属性和类别编码条件概率，由此定义由子集 $\Psi^1=\{S_i=\{X,C\}\forall X\in A\}$ 表示的具有复杂度的二维表格。其中三个 MapReduce 的输出情况如表 2~4 所示。表中的 $label$ 指类标签即 C_j，$token$ 是特征词，即 w_k。第一个 MapReduce 统计训练集中出现 w_k 的次数，计算每类中每个特征词的词频(TF)值：第二个 MapReduce 根据表 2 的输出文件，计算每个特征词的词频逆向文件频率(TFIDF)值。

待计算完毕后，将自动删除第一个 MapReduce 得出的 featureCount、wordFrep、termDocCount 三个文件。第四个 MapReduce 按照公式：$\sum\log[(TFIDF+1.0)/\pi(sigma_k+VocabCount)]$ 对表 3 的两个文件进行计算，并输出结果。

表 2　第一个 MapReduce 的输出情况

filename	key	value
wordFrep	$_WT,label,token$	类 C_j 中特征词 w_k 的 TF 值
termDocCount	$_DF,label,token$	类 C_j 中出现 w_k 的文档数
featureCount	$_FC,token$	训练集中出现 w_k 的文档总数
docCount	$_LC$	训练集中的文档总数

表 3　第二个 MapReduce 的输出情况

filename	输出 filename	key	value
1	Trainer-tfldf	$_WT,label,token$	TFIDF 值
2	Trainer-vocabCount	$_FS$	特征词总和

表 4　第三个 MapReduce 的输出情况

filename	输出 filename	key	value
1	$Sigma_j$	$_SJ,token$	每个特征词的 TFIDF 数量
2	$Sigma_k$	$_SJ,label$	每类中各特征词 TFIDF 总和
3	$Sigma_kSigma_j$	$_SJSK$	特征词的 TFIDF 总数

在之前的计算基础之上，最后 mapper 的返回值是测试文档则属于 C_j 类，与在其他类下的值进行比较，取出最大值所属的类标签值。

4.3　结果分析

实验数据将采用来源于 UCI KDD Archived 的 20 个 Newgroups 数据集[18~20]。在训练时，本实验使用了 Newgroups 数据集中所有类的文档。在测试时随机抽取了 politics、basetball、religion、hardware 和 motorcycle 共五类新闻数据组。这些真实的数据集提供了关于可伸缩性的不同属性，总结如表 5 所示。

表 5　实验中包含的真实数据集的属性

名称	属性数(n)	示例数(m)	大小/ GB
politics	134	50M	1.22
basetball	235	23M	2.11
religion	256	12M	0.97
hardware	631	4M	5.26
motorcycle	2000	500K	1.90

分类结果对比如表 6 所示。本文方法在所有五类新闻数据组上的分类准确性都取得了 99%以上的结果，并且均高与对比算法。实验体现了本文提出的基于 MapReduce 和 Apache Spark 框架的分布式朴素贝叶斯文本分类方法的准确性。

表 6　分类结果对比

类标签	测试文档数/篇	正确分类文档数/篇		
		文献[9]	文献[10]	本文方法
politics	234	218(93.16%)	229(97.86%)	234(100%)
basetball	256	246(96.09%)	251(98.05%)	255(99.61%)
religion	167	155(92.81%)	156(93.41%)	166(99.40%)
hardware	212	201(94.81%)	206(97.17%)	211(99.53%)
motorcycle	145	141(97.24%)	142(97.93%)	145(100%)

5　结束语

本文的重点是大数据技术，如 MapReduce，提出了一个适合当前云计算和高性能分布式编程范例的解决方案，还分析了最流行的 BNC 模型及其相应的学习算法。本文引入了一个通用的计算框架，可以通过实例化来学习任何所考虑的模型，并通过将大量数据用大量示例或属性作为目标，从而在广泛的问题上增加弹性和可伸缩性。还扩展了本文的建议，提出了不同的计算集群分布的策略。在 Newgroups 数据集进行实验，与对比方法相比，本文方法取得了比较优异的结果，证明了本文方

法的准确性。

本文还从理论和实践两个方面讨论了这些模型的最佳特性和主要缺陷，其目的在于找出未来的研究路线，以专门设计的算法来解决这个问题。

参考文献：

[1] 顾玉萍，程龙生. 基于 MTS-AdaBoost 的不平衡数据分类研究 [J]. 计算机应用研究，2018，35 (2): 346-348. (Gu Yuping, Cheng Longsheng. Classification of unbalanced data based on MTS-AdaBoost [J]. Application Research of Computers, 2018, 35 (2): 346-348.)

[2] 张继福，李永红，秦啸，等. 基于 MapReduce 与相关子空间的局部离群数据挖掘算法 [J]. 软件学报，2015，26 (5): 1079-1095. (Zhang Jifu, Li Yonghong, Qin Xizo, *et al.* Related-subspace-based local outlier detection algorithm using MapReduce [J]. Journal of Software, 2015, 26 (5): 1079-1095.)

[3] 黄廷辉，王玉良，汪振，等. 基于 Spark 的分布式交通流数据预测系统 [J]. 计算机应用研究，2018，35 (2): 405-409. (Huang Tinghui, Wang Yuliang, Wang Zhen, *et al.* Distributed traffic flow data prediction system based on Spark [J]. Application Research of Computers, 2018, 35 (2): 405-409.)

[4] 党红恩，赵尔平，刘炜，等. 利用数据变换与并行运算的闭频繁项集挖掘方法 [J]. 湘潭大学自然科学学报，2018，40 (1): 119-122. (Dang Hongen, Zhao Erping, Liu Wei, *et al.* Closed frequent item set mining base on data transformation and parallel computing [J]. Natural Science Journal of Xiangtan University, 2018, 40 (1): 119-122.)

[5] Baccarelli E, Cordeschi N, Mei A, *et al.* Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study [J]. Computers & Chemical Engineering, 2016, 91 (2): 182-194.

[6] Zeydan E, Bastug E, Bennis M, *et al.* Big data caching for networking: moving from cloud to edge [J]. IEEE Communications Magazine, 2016, 54 (9): 36-42.

[7] Sadeghi H, Valaee S, Shirani S. A weighted KNN epipolar geometry-based approach for vision-based indoor localization using smartphone cameras [C]// Proc of IEEE Sensor Array and Multichannel Signal Processing Workshop. [S. l.] : IEEE Press, 2014: 37-40.

[8] Hashem I A, Anuar N B, Gani A, *et al.* MapReduce: review and open challenges [J]. Scientometrics, 2016, 109 (1): 389-422.

[9] Kalayeh M M, Idrees H, Shah M. NMF-KNN: image annotation using weighted multi-view non-negative matrix factorization [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. [S. l.] : IEEE Computer Society, 2014: 184-191.

[10] Sun Kai, Kang H, Park H H. Tagging and classifying facial images in cloud environments based on KNN using MapReduce [J]. Optik-International Journal for Light and Electron Optics, 2015, 126 (21): 3227-3233.

[11] Hyunseok C, Kodialam M, Kompella R R, *et al.* Scheduling in mapreduce-like systems for fast completion time [C]// Proc of INFOCOM. [S. l.] : IEEE Press, 2015: 3074-3082.

[12] Chen Rong, Chen Haibo, Zang Binyu. Tiled-MapReduce: optimizing resource usages of data-parallel applications on multicore with tiling [C]// Proc of International Conference on Parallel Architectures and Compilation Techniques. [S. l.] : IEEE Press, 2017: 523-534.

[13] Yigitbasi N, Willke T L, Liao G, *et al.* Towards machine learning-based auto-tuning of MapReduce [C]// Proc of IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. [S. l.] : IEEE Press, 2014: 11-20.

[14] Murthy A C, Vavilapalli V K, Eadline D, *et al.* Apache Hadoop YARN: moving beyond MapReduce and batch processing with apache Hadoop 2 [Z]. 2014.

[15] 王卓，陈群，李战怀，等. 基于增量式分区策略的 MapReduce 数据均衡方法 [J]. 计算机学报，2016，39 (1): 19-35. (Wang Zhuo, Chen Qun, Li Zhanhuai, *et al.* An incremental partitioning strategy for data balance on MapReduce [J]. Chinese Journal of Computers, 2016, 39 (1): 19-35.)

[16] 徐德智，刘扬，Sarfraz Ahmed. 基于 Hadoop 的 RDF 数据存储及查询优化 [J]. 计算机应用研究，2017，34 (2): 477-480. (Xu Dezhi, Liu Yang, Sarfraz Ahmed. Optimization of RDF data storage and query based on Hadoop [J]. Application Research of Computers, 2017, 34 (2): 477-480, 486.)

[17] Yeager D S, Wang R. Comparing the accuracy of RDD telephone surveys and Internet surveys conducted with probability and non-probability samples [J]. Public Opinion Quarterly, 2015, 75 (4): 709-747.

[18] 齐芳，冯晰，徐其江. 基于人工鱼群优化的直推式支持向量机分类算法 [J]. 计算机应用与软件，2013，30 (3): 294-296. (Qi Fang, Feng Xi, Xu Qijiang. Direct push support vector machine classification algorithm based on artificial fish swarm optimization [J]. Computer Applications and Software, 2013, 30 (3): 294-296.)

[19] 池云仙，赵书良，罗燕，等. 基于特征隶属度的文本分类相似性度量方法 [J]. 计算机科学，2017, 44 (11): 289-296. (Chi Yunxian, Zhao Shuliang, Luo Yan, *et al.* Similarity measure for text classification based on feature subjection degree [J]. Computer Science, 2017, 44 (11): 289-296.)

[20] Trinh A T, Khambalia A, Ampt A, *et al.* Episiotomy rate in Vietnamese-born women in Australia: support for a change in obstetric practice in Viet Nam [J]. Bulletin of the World Health Organization, 2013, 91 (5): 350-356.